

A LEARNING CONTEXT – BASED E-RESOURCE RECOMMENDATION SYSTEM

Dr. Vivek Kumar Singh

Asst. Prof., Dept., of Computer Science, Banaras Hindu University, Varanasi – 221 005, India.

ARTICLE INFO

Article History:

Received: 19 Sep 2015;

Received in revised form:
22 Sep 2015;

Accepted: 22 Sep 2015;

Published online: 30 Sep 2015.

Key words:

Information Extraction,
Machine Reading,
RDF Schemas.

ABSTRACT

This paper presents experimental work to design a content-based recommendation system for eBook readers. The system automatically identifies a set of relevant eResources for a reader, reading a particular eBook, and presents them to the user through an integrated interface. The system involves two different phases. In the first phase, the textual content of the eBook currently read by the user is parsed to identify learning concepts being pursued. This requires analysing the text of relevant part(s) of the eBook to extract concepts and subsequently filter them to identify learning concepts of interest to Computer Science domain. In the second phase, a set of relevant eResources from the World Wide Web are identified and presented to reader. This involves invoking publicly available APIs from Slideshare, LinkedIn, YouTube etc. to retrieve relevant eResources for the learning concepts identified in the first part. The system is evaluated through a multi-faceted process involving tasks like sentiment analysis of user reviews of the retrieved set of eResources for recommendations.

Copyright © 2015 IJASRD. This is an open access article distributed under the Creative Common Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

In the era of massive outburst of new technologies coming into forefront every now and then, there has been a substantial increase in the electronic ocean of learning resources which not only help and guide the users but also make them lost in navigating this huge ocean of knowledge. In this regard, some kind of automation is required that could serve as a means of providing smooth passage through the navigational pathways of interlinked World Wide Web and suggest only user-specific or context-specific resources to the users.

This paper describes the experimental work carried out for designing such type of recommendation system that can show essential improvement in the learning outcome by supplementing concept-based set of knowledge resources with the learning environment for the concepts described in eBook. This system will be called as a content-based recommender

How to cite this article: Singh, V. K., & Khanka, S. S. (2015). "A Learning Context – Based E-Resource Recommendation System". *International Journal of Advanced Scientific Research & Development (IJASRD)*, 02 (03/II), pp. 114 – 121.

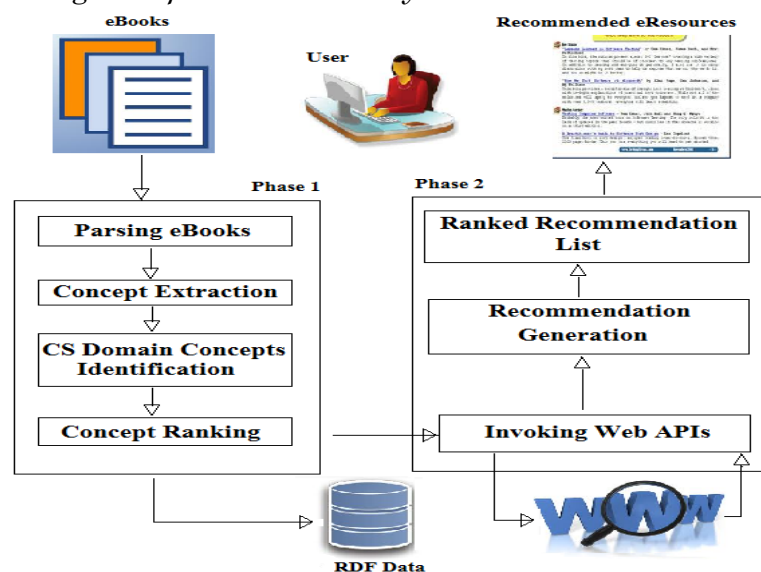
system because it works on concepts which are nothing but free form text and therefore a text analytics approach has been used to automatically identify those learning concepts which are being sought after by a user, reading an eBook. The system so designed contains two identifiable components, which work in two phases. First phase implements concept extraction from the parsed text of the eBook section being read by the reader using a text analytics based formulation. These extracted concepts are then filtered and ranked, and finally fed as input to the second phase. The second phase identifies the eResources and calls different APIs. The output of second phase is a ranked list that is presented to the user. The various eResources may include audios, videos, slides, documents, web articles, and twitter and linkedin ids of professionals working in the area. The block diagram of the proposed system, has been presented in figure 1 and it illustrates the system architecture, its components and an overview of its functioning.

The rest of the paper is organized as follows. Section 2 describes the architecture of recommender system and the step-wise working inside the system. Section 3 discusses the dataset used for this research work. Section 4 presents the experimental results. In Section 5, the paper concludes with a short summary of usefulness of this work and future extension.

ARCHITECTURE OF RECOMMENDER SYSTEM

The first phase of the recommendation system extracts learning concepts described a particular part of the eBook. This requires a number of tasks ranging from POS tagging to concept filtering. First task is parsing the textual contents of an eBook part and then use knowledge from linguistics to identify patterns that can represent concepts. The concepts so identified are subjected to a filtering process for identifying Computer Science (CS) domain concepts. The CS domain concepts present in a section are then ranked in order of importance for use by the recommendation generation phase. For concept extraction, first multitude of text extractions from the eBook is done, including extracting Table of Contents, Chapter and section texts. This was followed by POS tagging and terminological noun phrase identification.

Figure – 1: Block Diagram of Recommender System



2.1 Concept Extraction

In this phase, the concepts are extracted from eBooks. The concepts mainly are noun phrases or combination of adjective, noun and preposition. Three patterns (P1, P2, and P3) are considered for determining terminological noun phrases. The first two of these are from (Justeson and Katz, 1995), and the third one was proposed in (Agrawal et. al., 2012). These three patterns can be expressed using regular expressions as:

$P1 = C^*N$

$P2 = (C^*NP)?(C^*N)$

$P3 = A^*N^+$

where N refers to a noun, P a preposition, A an adjective, and $C = A|N$. The pattern P1 corresponds to a sequence of zero or more adjectives or nouns which ends with a noun and P2 is a relaxation of P1 that also permits two such patterns separated by a preposition. P3 corresponds to a sequence of zero or more adjectives, followed by one or more nouns. This pattern is a restricted version of P1, where an adjective occurring between two nouns is not allowed. The motivation for this pattern comes from sentences such as “The experiment with Swadeshi gave Mahatma Gandhi important ideas about using cloth as a symbolic weapon against British rule”. As a result of allowing arbitrary order of adjectives and nouns, “Mahatma Gandhi important ideas” is detected as a terminological noun phrase by pattern P1. But pattern P3 would result in the better phrases, “Mahatma Gandhi” and “important ideas”. Candidate concepts always comprise of maximal pattern matches. So there is no scope to have “density function” as a candidate in the presence of “probability density function”. The primary goal is to get greater specific concepts than general concepts. The similar kind of method was used in (Lent et al., 1997). In (Agrawal et. al., 2010), it is reported that the pattern P1 perform better than P2. The P3 pattern illustrates considerably better performance than P1.

2.2 CS Domain Concept Identification

The terminological noun phrases extracted represent generic noun-phrase based concepts. Not all of them represent concepts belonging to CS domain. In order to identify relevant eResources to recommend, it should be precisely known that what CS domain concepts are described in an eBook section. Therefore an attempt has been made to filter out the concepts that are not in the CS domain. For this, a filtering list has been used, which contains key concepts in CS domain. This list could not be an exhaustive list of CS domain concepts. This may result in losing some CS domain concepts, however, the list is appropriate enough to identify key concepts in different subjects of study in CS domain. ACM Computing Curricular Framework document (ACM CCF) is used as the base CS domain concept list. This list is augmented by incorporating it in terms from IEEE Computer Society Taxonomy and ACM Computing Classification System⁴. The augmenting process involved merging the two later documents into the first one, while preserving the 14 categories it is divided into. The combined list is thus a set of 14 different sets of CS domain knowledge areas, each knowledge area containing key concepts (the important ones) worth learning in that area. This list of concepts is used as the filtering list.

Every concept identified through the terminological noun phrase identification process, is subject to this filtering. However, exact term matching is still an issue. For example, two terms “algorithm complexity” and “complexity of algorithm” will not be a

match, if one goes for exact matching scheme. Therefore, for this case Jackard similarity measure has been used, which allows two concept phrases to result in a match even when the word order in the two is different, or there is an impartial match. The Jackard similarity equation is given in the equation below:

$$\text{Similarity}(A,B) = |A \cap B| / |A \cup B|$$

Where A and B stand for the two phrases. Here, $A \cap B$ is the set of common words in both phrases, $A \cup B$ is the set of union of words in both phrases and $|S|$ stands for the number of elements in the set S. A threshold value has to be set for deciding whether phrase A and B constitute a match. Empirically a threshold is found between 0.5 and 0.6, works best for identifying CS domain concepts. A simple example could help in understanding the suitability of this threshold. Consider, a phrase A= “methods of numerical analysis” is an identified terminological noun phrase and a phrase B=“numerical analysis methods” is a concept in the CS domain concept list. In this case the similarity score = 0.75, greater than threshold and confirming that A is a valid CS domain concept. Thus, by using the reference list and similarity scores, decisions will be made about every terminological noun phrase extracted from an eBook for being a valid CS domain concept.

2.3 Ranking of Concepts

Statistical measures of term occurrence have been used in the concerned section and the entire eBook to rank the concept(s) in order of importance. The rank score (section-rank) of a concept C_i belonging to a particular section S_j is computed as follows:

$$\text{RankScore}(C_i, S_j) = \text{Freq}(C_i, S_j) + \log_{10}(\text{NOC} / (\text{GRank}(C_i))) \quad (4)$$

where, $\text{Freq}()$ gives the number of occurrences of a particular concept in a given section, NOC refers to the total number of CS domain concepts extracted from the eBook and GRank is the rank of a concept in the entire eBook (with highest occurring concept getting the rank 1). Thus, there are two ranks for each concept C, a section-rank and a global-rank. The equation makes it clear that section-rank of a concept is computed by combining its occurrence measures in the section and the entire eBook. If the concept C_i refers to the highest ranking concept (rank 1), the $\text{Freq}(C_i, S_j)$ value is incremented substantially by addition of log normalized measure of its importance in the entire eBook. On the other hand, if the concept C_i refers to the concept with lowest global rank (rank=no. of concepts), its log normalized measure value becomes zero (since rank is equal to the number of concepts in eBook) and the section-rank of this concept is only a measure of its occurrence in the concerned section. In this manner, the importance of a concept will be computed in a given section (measured as section-rank). This is in a sense equivalent to attempting to find the key section (most important) for a concept (Agrawal et. al., 2010).

2.4 Generation of RDF

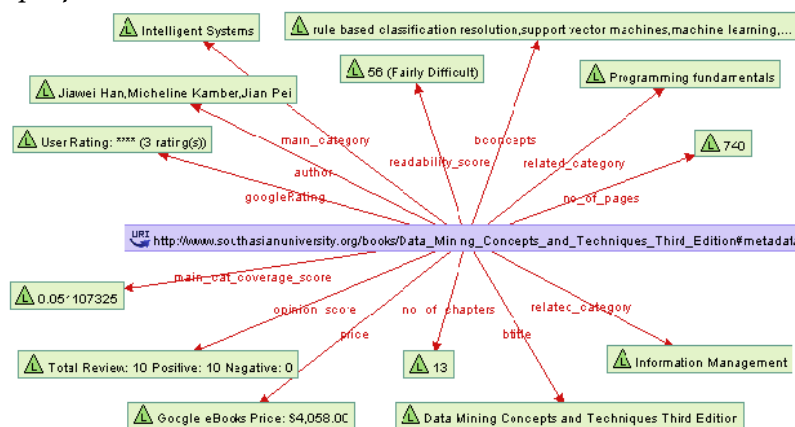
All this information is generated and written automatically (through program) in the RDF schema. The RDF schema contains `rdfs:resources` for the eBook metadata, concepts in a section and chapter, concept relations and eBook reviews obtained by crawling the Web. The eBook metadata comprises of eBook title, author, number of chapters, number of pages, eBook price, eBook rating, its main and two related categories as determined from augmented ACM CCF, coverage score, readability score and consolidated sentiment score profile. For each chapter node in the RDF, the entry consists of section and chapter titles,

top concepts with ranks, and relations extracted for the chapter. A sample example of RDF representation of eBook metadata is as follows:

```
<rdf:RDF
  xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:book="http://www.textanalytics.in/ebooks/
    Data_Mining_Concepts_and_Techniques_Third_Edition#" >
  <rdf:Description rdf:about="http://www.textanalytics.in/ebooks/
    Data_Mining_Concepts_and_Techniques_Third_Edition#metadata">
  <book:bttitle>Data Mining Concepts and Techniques Third Edition
  </book:bttitle>
  <book:author>JiaweiHan,MichelineKamber,Jian Pei</book:author>
  <book:no_of_chapters>13</book:no_of_chapters>
  <book:no_of_pages>740</book:no_of_pages>
  <book:bconcepts>rule based classification, resolution, support vector
    machines,machine learning,...</book:bconcepts>
  <book:main_category>Intelligent Systems</book:main_category>
  <book:main_cat_coverage_score>0.051107325</book:main_cat_coverage_score>
  <book:related_category>Programming fundamen-tals</book:related_category>
  <book:related_category>Information Management</book:related_category>
  <book:googleRating>User Rating: **** (3 rating(s))</book:googleRating>
  <book:readability_score>56 (Fairly Difficult)
  </book:readability_score>
</rdf:Description>
```

In this representation, the category and related category refers to the two closest of the 14 classes defined in ACM CCF. Similarly, other important information include readability score, author(s), number of pages etc. The figure 1 shows the RDF Graph for a part of the eBook metadata.

Figure – 2: RDF Graph for Book Metadata



2.5 Generating eResource Recommendation

After identifying important learning concepts presented in a section of eBook, the second phase of the system does generation of recommendations for relevant eResources for the learning concepts being pursued. While a section is being pursued by a reader, the key concepts in that section are identified and ranked. The top three concepts then form input

for the recommendation generation process. The design of the second part is fairly simple. First of all, useful eResources are explored for what may be readily available. Thereafter, a JAVA code has been implemented to invoke search APIs available for this purpose and integrate the results obtained. The system returns a number of eResources, slides from Slideshare5, web articles from Google Web Search, videos from YouTube, microblog posts in the area from Twitter, details of professionals working in the area from LinkedIn and related documents from DocStoc.

DATASET

The experimental evaluation has been performed on a moderate sized dataset of 30 eBooks from variety of sources in the domain of Computer science. The text corresponding to various parts of a PDF eBook is extracted using the iText API and programmatically reading the bookmarks. The different parts of an eBook are then parsed at a sentence level, starting with POS tagging and culminating in identification of C (denoted by terminological noun phrases).

RESULTS

In the following paragraphs snapshot of some results produced at various stages of processing by the system are presented. The snapshot of results shown correspond to a popular eBook on "Data Mining" that describes concepts and techniques of data mining and is a recommended eBook for graduate and research students. During phase 1 of system operation, all probable learning concepts (measured as terminological noun phrases) from a section of the eBook are extracted. Then these concepts are filtered using the augmented ACM CCF reference document. Some example CS domain concepts from beginning portion of this chapter are: business intelligence, knowledge management, entity relationship, models, information technology, database management system.

The second phase involves generation of recommendations for eResources relevant to the most significant learning concepts being pursued by the reader. The recommendations list contain eResources of various kinds. The recommendation list generated by us include videos from YouTube, slides from Slideshare, documents from DocStoc, Web articles from Google Web search, profile ids of professionals working in the area from LinkedIn and some others. A sample set of results for a concept "Data mining" from the first chapter of the eBook is presented below as an example demonstration. An example of recommended videos from YouTube for the concept are as follows:

Result for Concept: Data Mining

1. Thumbnail: <http://i.ytimg.com/vi/UzxYlbK2c7E/hqdefault.jpg>
URL: <http://www.youtube.com/watch?v=UzxYlbK2c7E>
2. Thumbnail: <http://i.ytimg.com/vi/EUzsy3W4I0g/hqdefault.jpg>
URL: <http://www.youtube.com/watch?v=EUzsy3W4I0g>

An example snapshot of recommended slides from SlideShare for the concept are as follows:

Result for Concept: Data Mining

1. Title: The Secrets of Building Realtime Big Data Systems

URL: <http://www.slideshare.net/nathanmarz/the-secrets-of-building-realtime-big-data-systems>

2. Title: Big Data with Not Only SQL

URL: <http://www.slideshare.net/PhilippeJulio/big-data-architecture>

A sample of recommended documents from DocStoc for the concept are as follows:

Result for Concept: Data Mining

1. Title: Data Mining

URL: <http://www.docstoc.com/docs/10961467/Data-Mining>

2. Title: Data Mining Introduction

URL: <http://www.docstoc.com/docs/10719897/Data-Mining-Introduction>

A snapshot of a part of recommended LinkedIn profiles for the concepts are as follows:

Result for Concept: Data Mining

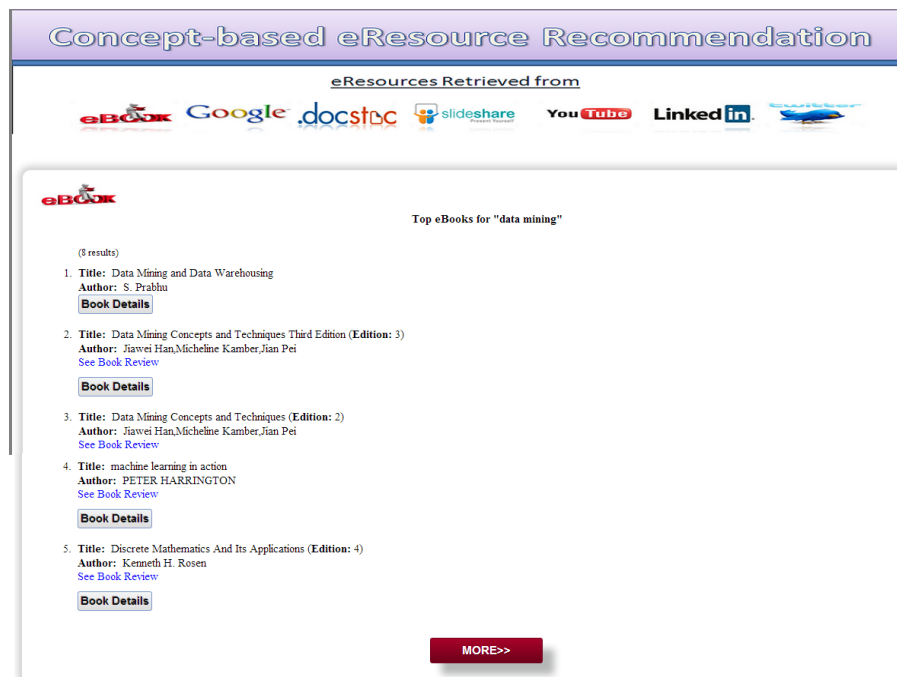
1. Name: Peter Norvig

URL: <http://www.linkedin.com/in/pnorvig?trk=skills>

2. Name: Daphne Koller

URL: <http://www.linkedin.com/pub/daphne-koller/20/3a8/405?trk=skills>

Figure – 3: *Recommended eBooks for Concept: Data Mining*



It would be important to mention here that the results displayed are a very small part of the actual results obtained. Through a similar process of API invocation, recommendations for top web links from Google Web Search and top profiles of persons writing on the topic on microblogging site Twitter are generated. The system thus generated recommendations for a comprehensive set of eResources (in addition to identifying the most relevant eBook and its chapter) for a concept being pursued by a learner.

For a given important concept in a section, recommend related eBooks (ranked in order of their relevance) are also recommended. The recommended list of related eBooks is

at present generated from the dataset collection itself. However, it is not a limitation and one can generate a list of related eBooks (related on the important learning concepts under consideration) from the Web. The list of related eBooks is ranked based on a computed sentiment score of their reviews obtained from Google book reviews and from Amazon. It was necessary to rank eBooks since the recommendation list of eBooks is not generated by an API having inherent ranking scheme, but by a concept-bases matching calculation. The purpose is that the most popular eBooks (measured through wisdom-of-crowds) should be ranked at top and recommended. For this, user reviews of all the eBooks in the dataset are collected by a selective crawling of Google Book review and Amazon sites.

The textual reviews obtained for each eBook are then labeled as 'positive' or 'negative' through a sentiment analysis program designed by us (Singh et. al., 2013a, 2013b). Thus for each candidate eBook, sentiment labels and strengths of its reviews (between 10-50 reviews) are computed and the strength scores are normalized (by dividing with number of 'positive' or 'negative' reviews) and used to rank the eBooks in order of their popularity. The figure 3 shows an example recommendation for the related eBooks recommended for concept "Data Mining".

CONCLUSION

The content-based recommendation system has been implemented for eBooks readers. The system takes eBook as an input and provides auxiliary resources for the the learning concepts to the user. The system utilizes text analytics techniques and work in two steps. In first step, it parse the eBook and, extract and identifies the important learning concepts. In next stages, it produces eResource recommendations which are more appropriate to the learning concepts and make available the users with additional learning resource material on the concept in concern. This system seems to be beneficial for a learners.

REFERENCES

- [1] Agrawal, R., Gollapudi, S., Kannan, A., & Kenthapadi, K. (2012). Data mining for improving textbooks. *ACM SIGKDD Explorations Newsletter*, 13(2), 7-19.
- [2] Agrawal, R., Gollapudi, S., Kenthapadi, K., Srivastava, N., & Velu, R. (2010, December). Enriching textbooks through data mining. In *Proceedings of the First ACM Symposium on Computing for Development* (p. 19). ACM.
- [3] Justeson, J. S., & Katz, S. M. (1995). Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural language engineering*, 1(01), 9-27.
- [4] Lent, B., Agrawal, R., & Srikant, R. (1997, August). Discovering Trends in Text Databases. In *KDD* (Vol. 97, pp. 227-230).
- [5] Singh, V. K., Piryani, R., Uddin, A., & Waila, P. (2013a, February). Sentiment analysis of Movie reviews and Blog posts. In *Advance Computing Conference (IACC), 2013 IEEE 3rd International* (pp. 893-898). IEEE.
- [6] Singh, V. K., Piryani, R., Uddin, A., & Waila, P. (2013b, March). Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification. In *Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on* (pp. 712-717). IEEE.